อาจหาญ สัตยารักษ์

สร้างโปรแกรม เรสซิเดนต์ (จบ) : เก็บภาพหน้าจอกราฟิก

เก็บรูปสวยบนจอใส่ลงดิสค์ เพื่อเพิ่มชีวิตชีวาให้... ผลงานที่สร้างสรรด้วยตนเอง

หลังจากตอนแรกเราได้โปรแกรมสำหรับพิมพ์ภาพหน้าจอลงกระดาษไปแล้ว ซึ่งคงช่วยให้การพิมพ์ภาพเป็นไปอย่างสะดวกยิ่งขึ้น เพราะว่าเมื่อต้องการพิมพ์ภาพใดก็เพียงแต่เปิดเครื่องพิมพ์และกดแป้นเพียงสองแป้นเท่านั้น จะได้ภาพบนกระดาษตามต้องการแล้ว

สำหรับคราวนี้จะมาเป็นภาพหน้าจอลงแผ่นบันทึก เพื่อนำไปแสดงประกอบกับโปรแกรมอื่น ๆ ที่ได้เขียนขึ้นในภายหลัง โดยซอฟต์เมล์ของตอนนี้มี 3 โปรแกรมหลักคือ

1. driver.pas สร้างยูนิตไฟล์สำหรับเข้าสู่กราฟิกโหมดโดยไม่ใช้ไฟล์ *.bgi

2. savescn.pas โปรแกรมเรสซิเดนต์สำหรับเก็บภาพหน้าจอ

3. showscn.pas โปรแกรมปกติสำหรับนำภาพที่เก็บด้วยโปรแกรมในข้อ 2 มาแสดงบนจอภาพ

นอกจากนี้จะมีไฟล์ประกอบอื่นคือ hercdrv.obj, Vgadrv.obj มาให้ด้วย

สร้างยูนิต DRIVER.TPU

เพื่อให้สามารถนำชุดคำสั่งการเข้ากราฟิกโดยไม่อาศัยไฟล์ *.bgi สามารถนำไปใช้ในโปรแกรมอื่น ๆ ได้โดยง่าย เราจึงจะสร้างยูนิตไฟล์ชื่อ driver.tpu ขึ้นมาใช้งาน เมื่อต้องการเข้าสู่ภาวะกราฟิกในไฟล์ใดก็เพียงเรียกใช้ยูนิตนี้เพียงยูนิตเดียว (ไม่ต้องมีไฟล์ *.obj อีกต่อไป)

การสร้างยูนิตไฟล์ทำได้ดังนี้

1. เขียนโปรแกรมตามปกติ

2. ที่บรรทัดแรกให้เขียนข้อความว่า UNIT ตามด้วยชื่อไฟล์ที่กำลังเขียน เช่น UNIT Driver ;

3. กำหนดส่วนที่เรียกใช้จากภายนอกยูนิตในขอบเขตของคำสั่ง interface ถึง implementation ซึ่งส่วนนี้อาจเป็นตัวแปรหรือชุดคำสั่งก็ได้ ถ้าเป็นตัวแปรต้องระวังไม่กำหนดตัวแปรนี้ซ้ำอีกในโปรแกรมที่เขียนขึ้นใหม่

4. ชุดคำสั่งในยูนิตไฟล์ที่สร้างนี้ไม่จำเป็นต้องเรียกใช้จากภายนอกก็ได้ กล่าวคือ อาจมีทั้งชุดคำสั่งที่ใช้อยู่ภายในยูนิตและชุดคำสั่งที่เรียกใช้ได้จากภายนอกยูนิต

5. ถ้าประสงค์จะให้ชุดคำสั่งใดทำงานเลยเมื่อเรียกใช้งานยูนิตก็กำหนดไว้ที่ส่วนของโปรแกรมหลักในยูนิตไฟล์ได้

6. เมื่อเขียนเสร็จก็ให้คอมพายล์-ไฟล์นี้เป็นไฟล์ยูนิตลงแผ่นบันทึกจะได้ driver.tpu มาใช้งาน การสร้างยูนิตไฟล์ driver.tpu ดังแสดงรูปที่ 1 โดยต้องเก็บไฟล์นี้ในชื่อ driver.pas เท่านั้น

มีรายละเอียดของโปรแกรมในรูปที่ 1 มีดังนี้

1. บรรทัดที่ 1-6 กำหนดชื่อยูนิต ยูนิตใช้งานร่วม ตัวแปรที่เรียกใช้จากภายนอกยูนิตและชุดคำสั่งที่เรียกได้จากภายนอก

2. บรรทัดที่ 7-10 ชุดคำสั่งนำไฟล์จัดการจอภาพสำหรับจอโมโนโครมและจอสีหรือจอวีจีเอมารวมไว้ในยูนิตไฟล์ Driver.tpu

3. บรรทัดที่ 11-25 การเข้ากราฟิกโหมดโดยไม่อาศัยไฟล์ *.bgi

4. ถ้าท่านต้องการเรียกใช้ชุดคำสั่ง opengraph; ในทันทีที่เรียกใช้ยูนิตให้ทำดังนี้

4.1 ตัดบรรทัดที่ 5 และ 25 ออกไป

4.2 เขียนคำสั่งต่อไปนี้แทนบรรทัดที่ 25 คือ

```
begin
  opengraph;
end.
```

ถ้าเขียนยูนิตไฟล์ตามข้อ 4 นี้ เมื่อเรียกใช้จะเข้าสู่กราฟิกโหมดโดยอัตโนมัติ

```
 1: UNIT Driver;  { DRIVER.PAS }
 2: interface
 3: uses dos,crt,graph;
 4: var VGA480:boolean;
 5: PROCEDURE Opengraph;
 6: implementation

 7: PROCEDURE HercDriver; external;
 8:   {$L Hercdrv.obj}
 9: PROCEDURE EgaVgaDriver; external;
10:   {$L Vgadrv.obj}

11: PROCEDURE Opengraph;
12: var grdriver,Graphck,Grmode:integer;
13:     Mono:boolean;
14: begin
15:    Mono:=false;
16:    detectgraph(grdriver,grmode);
17:    case grdriver of
18:      7:Mono:=true;
19:      9:If Pred(VGA480) then grmode:=1 Else grmode:=2;
20:    end;
21:  if Mono then Graphck:=RegisterBgidriver(@HercDriver)
22:         else  Graphck:=RegisterBgidriver(@EgavgaDriver);
23:   initgraph(grdriver,grmode,'');
24: end;
25: end.
```

รูปที่ 1 ยูนิต DRIVER.TPU สำหรับเข้าสู่กราฟิกโดยไม่อาศัยไฟล์ *.BGI

การเก็บภาพหน้าจอ

เมื่อเราพบเห็นภาพกราฟิกสวย ๆ จากโปรแกรมอื่น สิ่งหนึ่งที่ทุกคนต้องการก็คือ อยากได้ภาพนั้น ๆ มาใช้ในโปรแกรมของตนเอง แม้ว่าจะมีโปรแกรมสำเร็จในการเก็บภาพมาได้แต่บางครั้งก็ไม่ตรงกับความต้องการของเรา ดังนั้นการเขียนโปรแกรมเรสซิเดนต์เพื่อเก็บภาพนั้น ๆ เองจึงเป็นสิ่งที่ตรงกับความต้องการมากกว่า

วันนี้ผู้อ่านจะพบความสมหวังในเรื่องนี้ เพราะผู้เขียนได้นำเทคนิคมาเสนอให้ท่านแล้ว เพียงแต่ท่านพัฒนาส่วนการเก็บภาพให้ตรงกับความต้องการเท่านั้นก็จะได้โปรแกรมเก็บภาพหน้าจอไว้ใช้งาน

โปรแกรมในการเก็บภาพหน้าจออาจเป็นดังรูปที่ 2 ซึ่งเป็นส่วนหนึ่งของไฟล์ savescn.pas

รายละเอียดของโปรแกรมในรูปที่ 2 มีดังนี้

1. บรรทัดที่ 6-15 ส่วนต้นการกำหนดหน่วยความจำ ตัวแปรที่จำเป็นต่อการใช้งาน

2. บรรทัดที่ 16-33 ชุดคำสั่งในการเก็บภาพหน้าจอลงแผ่นบันทึก ซึ่งหลักการทำงานดังนี้

2.1 บรรทัดที่ 19-20 กำหนดชื่อไฟล์ที่จะเก็บภาพ โดยจะเริ่มที่ชื่อ screen1.pic ตามค่าของตัวแปร NO ซึ่งจะเพิ่มขึ้นจาก 1 เป็น 2, 3 จนถึง 99

2.2 บรรทัดที่ 21-22 กำหนด

```
 1: PROGRAM Print_Graphic_Screen;   { SAVESCN.PAS }
 2:  { Rasident Program
 3:    Alt+left_Shift      Save Graphic Page 0
 4:    Alt+Right_Shift     Save Graphic Page 1
 5:    Ctrl+Shift          Quit Rasident Program }

 6: {$M 3400,0,9000}
 7: uses dos,crt,graph,Driver;
 8: type Picdata=^PicD;
 9:      PicD=array[1..326] of byte;
10: var kbdFlag:byte Absolute $0040:$0017;
11:     Int9Save:Pointer; No:byte;
12:     Regs:registers;   ch:char;
13:     Drive,St,Nost:string;
14:     Pic:PicData;
15:     f:file;       Max,k:word;

16: PROCEDURE SaveGraphic;
17: var head:string;
18: begin
19:   str(No,NoST); Inc(No);
20:   St:='SCREEN'+NoST+'.pic';
21:   assign(f,ST);
22:   rewrite(f,1);  reset(f,1);
23:   head:='SCREEN*.PIC By Arthan Sattayarak';
24:   Blockwrite(f,Head,33);
25:   ch:=#26; Blockwrite(f,ch,1);
26:   Blockwrite(f,Max,2);
27:  for k:=0 to Max do
28:   begin
29:    getimage(0,k,getmaxx,k,pic^);
30:    blockwrite(f,Pic^,326);
31:   end;
32:  close(f);
33: end;
```

รูปที่ 2 ส่วนของการกำหนดยูนิตและตัวแปรใช้งาน พร้อมชุดคำสั่งเก็บภาพหน้าจอลงแผ่นบันทึก

ไฟล์ในหน่วยความจำ เพื่อเขียนลงแผ่นบันทึกตามชื่อไฟล์ที่หาได้จากบรรทัดที่ 20

2.3 บรรทัดที่ 23-24 กำหนดส่วนหัวของไฟล์เพื่อให้ทราบว่าไฟล์นี้เป็นไฟล์อะไร โดยส่วนนี้จะนำไปใช้ตรวจสอบเมื่ออ่านไฟล์ภาพเข้ามาว่าเป็นภาพที่กำหนดไว้หรือไม่

2.4 บรรทัดที่ 25 เขียนรหัสสิ้นสุดไฟล์ (EOF) เพื่อหลอกดอสให้คิดว่าไฟล์มีจุดสิ้นสุดเพียงแค่นี้ ซึ่งจะมีผลให้ใช้คำสั่ง type ไฟล์นี้จะปรากฏข้อความบนจอภาพด้วยข้อความที่กำหนดในบรรทัดที่ 23 เท่านั้น

2.5 บรรทัดที่ 26 เขียนจำนวนแถวสูงสุดของภาพที่เก็บไว้ (ตามค่าตัวแปร max) ซึ่งจะคงค่านี้ไว้ในโปรแกรมหลักที่อยู่ส่วนท้ายของไฟล์ savescn.pas นี้

2.6 บรรทัดที่ 27-32 ส่วนการเก็บภาพลงแผ่นบันทึก ใช้วิธีการจำภาพครั้งละหนึ่งแถวตามแนวนอนนับจากบนลงล่าง โดยการใช้คำสั่ง getimage จำภาพลงในตัวแปร pic แล้วเขียนค่าทั้งหมดนี้ลงในแผ่นบันทึกในไฟล์ที่กำหนดไว้ เมื่อได้ภาพครบถ้วนก็ปิดไฟล์

การเก็บภาพหน้าจอลงในไฟล์ screen*.pic นี้ เป็นการเก็บไฟล์ลงในพาธที่กำลังติดต่ออยู่ หากได้เปลี่ยนพาธไปก็จะเปลี่ยนที่เก็บภาพไปเรื่อย ๆ ด้วย ดังนั้นถ้าท่านประสงค์จะให้เก็บลงที่พาธคงที่ใดคงที่หนึ่งก็ให้เพิ่มคำสั่งในบรรทัดที่ 20 ให้เพิ่มชื่อพาธเข้าไปข้างหน้าชื่อไฟล์ด้วย เช่น

```
ST:='C:\DATAPIC\SCREEN'+NosT+'.PIC';
```

ผู้อ่านอาจกำหนดเป็นพารามิเตอร์ชื่อพาธเข้ามาด้วยก็ได้ ทั้งนี้ต้องแก้ไขโปรแกรมเอาเอง

ชุดคำสั่งตรวจการกดแป้น

เพื่อให้การทำงานเป็นไปโดยอัตโนมัติเมื่อตรวจพบการกดแป้นพิมพ์ที่ต้องการ เราจำเป็นต้องเขียนส่วนการตรวจการกดแป้นพิมพ์นี้ ให้ตรงกับความต้องการซึ่งจะมีดังนี้

ถ้ากดแป้น Alt และ Left Shift พร้อมกันจะเป็นการเก็บภาพกราฟิกหน้า 0

ถ้ากดแป้น Alt และ Right Shift พร้อมกันจะเป็นการเก็บภาพกราฟิกหน้า 1

ถ้ากดแป้น Ctrl และ Shift (ซ้ายหรือขวาก็ได้) พร้อมกันจะหยุดการ

ทำงานของโปรแกรม

เมื่อนำมาเขียนโปรแกรมจะได้ชุดคำสั่งดังรูปที่ 3 ซึ่งหวังว่าผู้อ่านคงทำความเข้าใจกับชุดคำสั่งต่าง ๆ ได้เอง เพราะบางส่วนก็นำจากตอนที่ 1 มาใช้

ส่วนของโปรแกรมหลัก

เพื่อให้ผู้เรียกใช้ไฟล์นี้ทราบถึงความสามารถในการทำงานและเงื่อนไขของการใช้แป้นพิมพ์ เราจึงต้องเขียนรายละเอียดให้ผู้ใช้ทราบเมื่อรันโปรแกรม พร้อมกำหนดทางเลือกให้โดยการป้อนพารามิเตอร์ให้ตรงกับความต้องการแสดงดังรูปที่ 4 ซึ่งรายละเอียดดังนี้

1. บรรทัดที่ 67 เก็บชื่อพาธที่รันไฟล์นี้ไว้ในตัวแปร Drive เพื่อนำไปใช้ในคำอธิบายการใช้งาน

2. บรรทัดที่ 68-81 แสดงคำอธิบายรายละเอียดการทำงานและเงื่อนไขการใช้แป้นต่าง ๆ ตลอดจนแจ้งให้ทราบถึงพารามิเตอร์ที่ใช้ประกอบ หากกดแป้น S หรือ s จะฝังโปรแกรมนี้ลงในหน่วยความจำให้เป็นโปรแกรมเรสซิเดนต์ โดยจะทำการเข้าสู่โหมดกราฟิกตามที่กำหนดไว้ พร้อมกับจะวนรอดักการกดแป้นตามที่เรากำหนดไว้

3. เมื่อกดแป้น S หรือ s ก็จะเป็นการทำงานตามบรรทัดที่ 83-95 ซึ่งมีรายละเอียดดังนี้

3.1 บรรทัดที่ 85-86 กำหนดตัวแปรแสดงลำดับไฟล์ (NO) และตรวจสอบว่าต้องการใช้กราฟิกขนาด 640×348 จุดหรือขนาด 640×480 จุด

3.2 บรรทัดที่ 87-89 เข้าสู่กราฟิกโหมด กำหนดค่าของตัวแปร Max กำหนดขอบเขตของจอภาพใช้งาน

3.3 บรรทัดที่ 90 จองหน่วยความจำฮีพเพื่อรองรับตัวแปร pic

```
34: PROCEDURE UpdateStatus;
35: begin
36:   if ((KbdFlag and 6)=6) or ((KbdFlag and 5)=5) then
37:    begin
38:      setIntVec(9,int9Save);
39:      dispose(Pic);
40:      closeGraph;
41:      halt;
42:    end;
43:   if (KbdFlag and 10)=10 then
44:    begin
45:      setactivepage(0);
46:      saveGraphic;
47:    end;
48:   if (KbdFlag and 9)=9 then
49:    begin
50:      setactivepage(1);
51:      SaveGraphic;
52:    end;
53: end;

54: {$F+}
55: PROCEDURE Int9Handler;
56: {$F-}
57: Interrupt;
58: begin
59:   Inline($9C/
60:          $3E/$FF/$1E/Int9Save/
61:          $FA);
62:     UpDateStatus;
63:   Inline($FB);
64: end;
```

รูปที่ 3 ส่วนการตรวจสอบการกดแป้นพิมพ์เพื่อเก็บภาพหน้าจอ

3.4 บรรทัดที่ 91-94 ทำการฝังโปรแกรมทั้งหมดลงในหน่วยความจำให้เป็นเรสซิเดนต์เพื่อนำไปใช้งานตามต้องการต่อไป

เนื่องจากโปรแกรมนี้เขียนด้วยภาษาชั้นสูง ดังนั้นโปรแกรมจะใช้หน่วยความจำมาก จึงอาจมีปัญหาในการรันไฟล์กราฟิกบางไฟล์ แต่เมื่อผู้เขียนได้ทดสอบกับไฟล์ที่เขียนด้วยภาษาปาสคาลและเกมบางเกมก็พบว่าสามารถรันโปรแกรมและเก็บภาพหน้าจอได้

การใช้งานไฟล์ SAVESCN.EXE

การใช้งานก็ทำดังนี้

1. รันไฟล์ savescn.exe ดังนี้ (สมมติอยู่ในไดร์ฟ A)

A>savescn	ถ้าต้องการเก็บภาพจากจอภาพกราฟิกขนาด 640×348 จุด
A>savescn 480	ถ้าต้องการเก็บภาพจากจอภาพกราฟิกขนาด 640×480 จุด

```
65: {====================== MAIN ====================}
66: begin
67:   getdir(0,drive);    Drive:=Drive+'>';
68:   clrscr;
69:   writeln('Rasident Program  for Save Graphics Screen');
70:   writeln('=== By ARTHAN SATTAYARAK Lampang 52000 ===');
71:   writeln;
72:   writeln('Alt+left_Shift      Save Graphic Page 0');
73:   writeln('Alt+Right_Shift     Save Graphic Page 1');
74:   writeln('Ctrl+Shift          Quit Rasident Program');
75:   writeln;
76:   writeln('     === Graphics 640x350 or 720x348 ===');
77:   writeln('usage : '+Drive+'savescn ');
78:   writeln('     === Graphics 640x480 (VGA only) ===');
79:   writeln('usage : '+Drive+'savescn 480 ');
80:   writeln;
81:   writeln('press  S  == for Install Program');
82:   ch:=readkey;
83:  if Upcase(ch)='S' then
84:   begin
85:     No:=1;  VGA480:=false;
86:     if Paramstr(1)='480' then VGA480:=true;
87:     Opengraph;  cleardevice;
88:     max:=getmaxy; if VGA480 then Max:=480;
89:     setviewport(0,0,getmaxx,Max,clipon);
90:     new(pic);
91:     UpDateStatus;
92:     getIntVec(9,Int9Save);
93:     setIntVec(9,@int9handler);
94:     keep(0);
95:  end;
96: end.
```

รูปที่ 4 ส่วนของโปรแกรมหลักแสดงรายละเอียดของโปรแกรมและพารามิเตอร์ที่ใช้งาน

เมื่อข้อความปรากฏบนจอก็ให้กดแป้น S หรือ s หรือแป้นอื่น

2. รันไฟล์ที่ให้ภาพกราฟิกบนจอภาพที่เราต้องการเก็บภาพจากจอภาพ

3. เมื่อภาพที่ต้องการปรากฏบนจอภาพให้ทำดังนี้

กดแป้น Alt+Left Shift เพื่อเก็บภาพกราฟิกหน้า 0 หรือกดแป้น Alt+Right Shift ให้เก็บภาพกราฟิกในหน้า 1 (โดยทั่ว ๆ ไปมักจะแสดงกราฟิกในหน้า 0) จะได้ไฟล์มีชื่อ screen*.pic โดยที่ * จะเริ่มที่เลข 1 และเพิ่มขึ้นเรื่อย ๆ จนถึงภาพลำดับที่ 99

ในโปรแกรมไม่ได้ป้องกันการเก็บภาพในกรณีที่เป็นภาพลำดับที่มากกว่า 99 ไว้ ทั้งนี้เพราะว่าคงไม่มีใครเก็บภาพหน้าจอมากขนาดนั้น

4. ถ้าต้องการยกเลิกการโปรแกรมเรสซิเดนต์ให้กดแป้น Alt+Shift การยกเลิกแบบนี้ไม่ได้ยกเลิกหน่วยความจำที่โปรแกรมฝังตัวอยู่ แต่เป็นการยกเลิกการทำงานของโปรแกรมเท่านั้น ดังนั้น เมื่อรันไฟล์นี้ภาพที่เก็บจะเริ่มที่ screen1.pic เสมอ

การเก็บภาพตามโปรแกรมจะใช้เนื้อที่ในแผ่นมาก เพราะจะได้ทั้งสีสรรของภาพมาด้วย หากเก็บภาพลงแผ่นบันทึกก็อาจได้แผ่นละไม่กี่ภาพ แต่เนื่องจากในยุคนี้เราจะใช้งานบนฮาร์ดดิสก์เป็นส่วนใหญ่ ดังนั้นก็อาจตัดปัญหาเรื่องขนาดของภาพไปได้

โปรแกรมนี้สามารถเก็บภาพหน้าจอจากภาพที่ใช้เครื่องสแกนเข้ามาได้ เพราะผู้เขียนได้ทำการทดลองเก็บภาพมาแล้ว ภาพที่เก็บนี้จะเป็นการเก็บทั้งจอภาพ ถ้าผู้อ่านมีความประสงค์จะตัดภาพให้เล็กลงในขนาดที่ต้องการ ก็คงต้องพัฒนาโปรแกรมตามปกติเพิ่มเติม โดยอาศัยไฟล์ showscn.pas เรียกภาพนั้น ๆ มาแสดงบนจอภาพแล้วเขียนโปรแกรมตัดต่อตามต้องการ

นำภาพที่เก็บไว้มาแสดงบนจอภาพ

เมื่อเราได้ภาพเก็บลงแผ่นบันทึกแล้ว ถ้าต้องการนำภาพนั้น ๆ มาแสดงบนจอภาพอีกครั้งหนึ่ง เราจะต้องเขียนโปรแกรมขึ้นมาอีกโปรแกรมหนึ่งโดยอาศัยหลักการเดียวกันกับการเก็บภาพไว้ ซึ่งตามตัวอย่างการเก็บนั้นเราคงต้องกำหนดการนำภาพมาอ่านได้ดังนี้

1. ตรวจสอบว่าไฟล์ที่เรียกมานี้เป็นไฟล์ภาพที่เรากำหนดหรือไม่

2. ถ้าไม่ใช่ไฟล์ภาพที่เราเก็บไว้ก็เขียนข้อความแจ้งให้ทราบว่าไม่ใช่แฟ้มภาพของเรา

3. ถ้าใช่ก็จะทำการอ่านขนาดความสูงของภาพ (ตัวแปร Max) เข้ามาเพื่อกำหนดการเข้ากราฟิกให้ตรงกับ

รูปที่ 5
โปรแกรมในการนำภาพที่เก็บไว้บนแผ่นบันทึกมาแสดงบนจอภาพ

```
 1: PROGRAM Show_Graphic_Screen;    { SHOWSCN.PAS  }
 2: uses dos,crt,graph,Driver;
 3: var
 4:     ch:char;
 5:     Max:word;
 6:     Picname,head,Filename:string;
 7:     f:file;

 8: PROCEDURE ShowGraphic;
 9: var st,NoST:string;
10:     Pic:array[1..326] of byte;
11:     k:word;
12: begin
13:   assign(f,PicName);
14:   reset(f,1);
15:   blockread(f,head,34);
16:   Blockread(f,Max,2);
17:  for k:=0 to Max do
18:   begin
19:    blockread(f,Pic,326);
20:    Putimage(0,k,pic,0);
21:   end;
22:   close(f);
23: end;

24: {=================== MAIN ===================}
25: begin
26:   clrscr;
27:   writeln('   '#17#17#17'PROGRAM Showpic for SAVESCN.EXE Only'#16#16#16);
28:   writeln('       By ARTHAN SATTAYARAK Lampang 52000');
29:   writeln('================================================');
30: if Paramstr(1)<>'' then
31:   begin
32:     Picname:=paramstr(1);
33:     if pos('.',PicName)=0 then Picname:=Picname+'.PIC';
34:     assign(f,PicName);
35:     {$I-}  reset(f,1); {$I+}
36:     if IOResult<>0 then
37:       begin
38:         writeln('file ',Picname,' not found');
39:         writeln;
40:         halt;
41:        end;
42:     blockread(f,head,33);
43:     if head<>'SCREEN*.PIC By Arthan Sattayarak' then
44:       begin
45:          close(f);
46:          writeln('file ',Picname,' Incorrect FORMAT');
47:          writeln;
48:          halt;
49:       end;
```

รูปที่ 5 (ต่อ)

```
50:     Blockread(f,ch,1);
51:     Blockread(f,Max,2);  close(f);
52:     VGA480:=False;  if max>349 then VGA480:=true;
53:     Opengraph;
54:     ShowGraphic;
55:     ch:=readkey;
56:     closegraph;
57:   end
58:   else
59:    begin
60:     getdir(0,FileName);
61:     FileName:=FileName+'>';
62:     writeln('usage :  ',FileName,'showscn nameofpicfile');
63:     writeln;
64:    end;
65: end.
```

ภาพที่เก็บ (ในกรณีที่ทราบขนาดภาพแน่นอนแล้ว จะเข้าสู่เข้ากราฟิกโดยไม่ตรวจส่วนนี้เลยก็ได้)

4. จากนั้นก็อ่านส่วนที่เป็นภาพเข้ามาสู่หน่วยความจำก็จะได้ภาพปรากฏบนจอภาพตามต้องการ นำมาเขียนโปรแกรมได้ดังรูปที่ 5 ซึ่งก็คือโปรแกรม showscn.pas ในซอฟต์เมล์

มีรายละเอียดดังนี้

1. บรรทัดที่ 1-7 กำหนดยูนิตใช้งานและตัวแปร

2. บรรทัดที่ 8-15 กำหนดไฟล์แล้วอ่านข้อมูลส่วนหัวทิ้ง

3. บรรทัดที่ 16 อ่านค่าความสูงของภาพที่เก็บมาใส่ในตัวแปร Max

4. บรรทัดที่ 17-21 อ่านข้อมูลภาพเข้ามาไว้ในตัวแปร pic แล้วลงภาพด้วยคำสั่ง putimage จนครบตามค่าตัวแปร Max

5. บรรทัดที่ 22 ปิดไฟล์และจบการแสดงภาพบนจอภาพ

6. บรรทัดที่ 25-29 ส่วนของโปรแกรมหลักแสดงให้ทราบเป็นโปรแกรมอะไร

7. บรรทัดที่ 30-37 เมื่อได้กำหนดชื่อไฟล์มาให้ด้วยก็จะทำงานตามขั้นตอนต่อไปนี้

7.1 บรรทัดที่ 32-35 กำหนดชื่อไฟล์ภาพจากพารามิเตอร์ตัวแรกที่กำหนดมาให้พร้อมกับการรันไฟล์ showscn.exe กำหนดไฟล์พร้อมกับตรวจสอบว่ามีไฟล์ตามที่กำหนดหรือไม่

7.2 บรรทัดที่ 36-41 ถ้าไม่พบไฟล์นี้ก็แสดงข้อความให้ทราบแล้วจบการทำงาน

7.3 บรรทัดที่ 42-49 อ่านส่วนหัวเพื่อตรวจสอบว่าใช้ไฟล์ภาพที่เก็บไว้หรือไม่ ถ้าไม่ใช่ก็จะมีข้อความแจ้งให้ทราบแล้วจบการทำงาน

7.4 บรรทัดที่ 50-52 อ่านค่าจากไฟล์ภาพเพื่อหาค่าความสูงของภาพ (ตัวแปร Max) เพื่อกำหนดว่าเป็นภาพกราฟิกขนาดใด

7.5 บรรทัดที่ 53-57 เข้าสู่ภาวะกราฟิกและแสดงภาพบนจอภาพ

8. หากไม่มีการป้อนชื่อไฟล์ที่จะอ่านก็จะแสดงข้อความให้ทราบตามคำสั่งบรรทัดที่ 59-64 การเรียกใช้โปรแกรมนี้ทำได้โดยการป้อนชื่อไฟล์ภาพ เช่น

```
A>showscn  screen1.pic
C>showscn  c:\data\pic\screen11.pic
เป็นต้น
```

ผลการทำงานของโปรแกรมทั้งสองจะเป็นอย่างไรก็คงต้องทดลองกันเอง

หวังว่าท่านคงสามารถจับภาพจากโปรแกรมอื่น ๆ นำมาใช้ในโปรแกรมของท่านได้ง่ายขึ้น ทั้งยังหวังอีกว่าคงจะมีบางท่านนำแนวทางนี้ไปพัฒนาเพื่อใช้งานให้ตรงกับความต้องการของตนเองได้มากยิ่งขึ้น พบกันใหม่คราวหน้า ...ด้วยเรื่องอื่น ๆ ที่น่าสนใจ